



# Named entity recognition

## Evaluation of Existing Systems

**Bowen Sun**

Master in Information Systems  
Submission date: July 2010  
Supervisor: Jon Atle Gulla, IDI

Norwegian University of Science and Technology  
Department of Computer and Information Science



# Problem Description

Named entity recognition (NER) is a technology for recognizing proper nouns (entities) in text and associating them with the appropriate types. Common types in NER systems are location, person name, date, address, etc. Some NER systems are incorporated into Parts-of-Speech (POS) taggers, though there are also many stand-alone applications. Whereas most NER systems are based on analyzing patterns of POS tags, they also often make use of lists of typed entities (like list of possible person names) or regular expressions for particular types (like address patterns). The purpose of this project is to evaluate NER systems for English and Norwegian. A number of available systems need to be tested and compared to each other. The evaluation should include a verification of which entity types that can be supported by the different systems. We also want to investigate to what extent lists of typed entities and/or regular expressions may be used by the systems.

If possible we want lists of typed entities and regular expressions to be prepared for and incorporated into a selected NER system

Assignment given: 03. February 2010

Supervisor: Jon Atle Gulla, IDI



# Abstract

Nowadays, one subfield of information extraction, Named Entity Recognition, becomes more and more important. It helps machine to recognize proper nouns (entities) in text and associating them with the appropriate types. Common types in NER systems are location, person name, date, address, etc. There are several NER systems in the world. What's the main core technology of these systems? Which kind of system is better? How to improve this technology in the future? This master thesis will show the basic and detail knowledge about NER.

Three existing NER systems will be choose to evaluate in this paper, GATE, CRFClassifier and LbjNerTagger. These systems are based different NER technology. They can stand for the most of NER existing systems in the world now. This paper will present and evaluate these three systems and try to find the advantage and disadvantage of each system.



# Preface

This report documents the work done in my master thesis at the Information Systems Group of the Department of Computer and Information Science, Faculty of Information Technology, Mathematics and Electrical Engineering at Norwegian University of Science and Technology.

I would like to thank my supervisor Professor Jon Atle Gulla for his guidance, feedback and comments during my work. He gives me very useful information and I would also like to thank my second supervisor Arne Dag Fidjestøl for his useful guidance of workbench and select corpus for my project.





# Contents

1. Introduction.....	1
1.1 Problem.....	1
1.2 Objectives.....	2
1.3 Assumptions.....	2
1.4 Results.....	3
1.5 Report Structure.....	4
2. Theory and Background.....	7
2.1 Ontologies.....	7
2.2 NER in Ontology Learning.....	8
2.3 Alternatives to NER.....	9
2.3.1 Information Extraction.....	10
2.4 Learning method of NER.....	10
2.4.1 Supervised Learning.....	11
2.4.2 Semi-Supervised Learning.....	12
2.4.3 Unsupervised Learning.....	15
3. Named Entity Recognition.....	17
3.1 Named Entity Recognition Systems.....	17
3.2 Named Entity types.....	19
3.3 Application of Named Entity Recognition.....	21
3.4 Feature space of NER.....	22
3.4.1 Word-level features.....	23

3.4.2	List Look up features.....	23
3.4.3	Document and corpus features.....	24
4.	Existing Systems.....	27
4.1	GATE.....	28
4.2	LbjNerTagger.....	34
4.3	CRFClassifier.....	36
5.	Evaluation of Existing Systems.....	41
5.1	Corpus for evaluation.....	41
5.1.1	Collection of corpus.....	41
5.1.2	Characterization of corpus.....	42
5.2	Evaluation Method.....	42
6.	Evaluation Results.....	47
6.1	Concepts.....	47
6.2	Evaluation results of GATE.....	48
6.3	Evaluation results of CRFClassifier.....	50
6.4	Evaluation results of LbjNerTagger.....	52
6.5	Evaluation Summary.....	53
7.	Conclusion.....	55
8.	Future Work.....	57
Appendix A: Acronyms and Abbreviations.....		59
Appendix B: Digital Appendix.....		61
Reference.....		63

# List of Figures

4.1 Screenshot of GATE.....	29
4.2 The pipeline of ANNIE components.....	32
4.3 Screenshot of LbjNerTagger.....	34
4.4 Screenshot of CRFClassifier.....	37
4.5 Textual Entailment Pipeline of CRF.....	39
5.1 Process of Evaluation.....	43
5.2 Definition of Recall and Precision.....	44
5.3 Interface of developed program.....	46



# List of Tables

4.1 Character of each system.....	28
4.2 Model Trades-off.....	39
4.3 Features of CRF for NER.....	40
5.1 Evaluation table.....	46
6.1 The key words should be test.....	48
6.2 Evaluation result of Organization for GATE.....	48
6.3 Evaluation result of Location for GATE.....	49
6.4 Evaluation result of Person for GATE.....	49
6.5 Evaluation result of Organization for CRFClassifier.....	50
6.6 Evaluation result of Location for CRFClassifier.....	51
6.7 Evaluation result of Person for CRFClassifier.....	51
6.8 Evaluation result of Organization for LbjNerTagger.....	52
6.9 Evaluation result of Location for LbjNerTagger.....	53
6.10 Evaluation result of Person for LbjNerTagger.....	53



# Chapter 1

## Introduction

At the Department of Computer and Information Science (IDI) at Norwegian University of Science and Technology (NTNU) there has been substantial work in the information retrieval. One of the researched areas is Named Entity Recognition. The goal of Named Entity Recognition is to identify and classify the proper names appearing in the text and the number of meaningful phrases. This master thesis is a part of the ongoing research in the field of information retrieval.

### 1.1 Problem

The following is a quote of the problem description:

*“Named entity recognition (NER) is a technology for recognizing proper nouns (entities) in text and associating them with the appropriate types. Common types in NER systems are location, person name, date, address, etc. Some NER systems are incorporated into Parts-of-Speech (POS) taggers, though there are also many stand-alone applications. Whereas most NER systems are based on analyzing patterns of POS tags, they also often make use of lists of typed entities*

*(like list of possible person names) or regular expressions for particular types (like address patterns).*

*The purpose of this project is to evaluate NER systems for English and Norwegian. A number of available systems need to be tested and compared to each other. The evaluation should include a verification of which entity types that can be supported by the different systems. We also want to investigate to what extent lists of typed entities and/or regular expressions may be used by the systems.*

*If possible we want lists of typed entities and regular expressions to be prepared for and incorporated into a selected NER system.”*

The main challenge of this problem is to find the suitable NER systems and try to find the difference of their Named entity types. Test the systems by using the right method, evaluate the results in the right way. This thesis will show the solution of this challenges.

## 1.2 Objectives

With the problem description as a basis, there are two main objectives been extracted:

- List and analyst the state of art of the Named Entity Recognition Systems
- Evaluate the results against existing systems

Through the developed program and test the corpus to evaluate the existing systems. According to the value of recall and precision, find the advantage and disadvantage of each NER system.

## 1.3 Assumptions



Based on the problem description and the objectives, the following assumptions are taken:

- Search, download, install and test the existing NER systems
- The corpus for evaluate the existing NER systems need to be provided
- A program need to be developed to test the document that have been processed by the existing NER systems

## 1.4 Result

The evaluation results of Named Entity Recognition shows in this thesis. Although there are several existing NER systems in the world, we only evaluate three of them, GATE, CRFClassifier and LbjNerTagger. These systems are typical of existing NER systems. Through the research and developed program, we can easily find that GATE is more diversification, CRFClassifier is more standardization and LbjNerTagger is average among them. The recall of every system is almost 100%. The precision of GATE has a tremendous difference between different Named entity types. The average of value of precision is about 60%. As to the some NEs, GATE hasn't recognized them very well. The precision of CRFClassifier is almost 70%. Compare with LbjNerTagger, it has the same value of precision. About tag sets, GATE has more kinds of tag sets, such as job titles, first name, etc. CRFClassifier and LbjNerTagger only have the person, location and organization. The tag that have recognized by GATE and CRFClassifier will show as <person>Jack</person>; LbjNerTagger will show it as [PER Jack ]. They are all very easy to identify the different types of Named Entities.

## 1.5 Report Structure

The structure of this report is as follows:

- Chapter 2: Theory and Background

This Chapter introduces background and theory that is important in the field of named entity recognition. Relevant techniques of information extraction and information retrieval

- Chapter 3: Named Entity Recognition

This chapter will introduces the basic idea and definition of Named Entity Recognition

- Chapter 4: Existing Systems

This chapter introduces the existing Named Entity Recognition Systems: GATE, LbjNerTagger and CRFClassifier.

- Chapter 5: Evaluation of Existing Systems

This chapter will present the chosen evaluation method for the evaluation of existing systems.

- Chapter 6: Evaluation Results

This chapter present and discusses the evaluation results

- Chapter 7: Conclusion

This chapter presents concluding remarks for the work done

- Chapter 8: Future Work

This chapter discusses possible direction for further work.

Appendices:

- Appendix A: Acronyms and Abbreviations
- Appendix B: Digital Appendix



## Chapter 2

# Theory and Background

At IDI of NTNU, substantial work has been made in the field of information retrieval. One of the researched areas are information extraction. Named Entity Recognition is part of the information extraction. It is known as entity identification and entity extraction. This chapter introduces background and theory that is important in the field of named entity recognition.

### 2.1 Ontologies

The concept of ontology, initially originated in philosophy, is an objective deposit in a system of explanations or statements concerned with the abstract nature of objective reality. In the area of artificial intelligence, an ontology defines the basic terms and relations comprising the vocabulary of a topic area as well as the rules for combining terms and relations to define extensions to the vocabulary [1]. And the most widely accepted definition of ontology is a "formal, explicit specification of a shared conceptualization" [2].

Nowadays, ontologies are used more and more popular in the field of computer science and information systems. Especially, the concept of semantic web, is strongly connected with ontologies. Tim Berners-Lee[3], who raises the concept of semantic web, said that the current Web is for people to read and understand, and it documents a growing media, is not conducive to the realization of the automated processing of data and information. A new generation of Semantic Web will not only human but also for the computer (information agents) to bring semantic content, so that the computer (or the information agent) can "understand" web content, so as to realize the automation of information processing. He believes that the current Web and the Semantic Web is not isolated from another Web, but rather the expansion of the current Web in the Semantic Web, information through well-defined semantics, better able to promote between the computer and mutual cooperation.

Nowadays, ontologies used widely in the world. First, using in the semantic search, which is a process used to improve online searching by using data from semantic networks to disambiguate queries and web text in order to generate more relevant results. Second, it can be used in interoperability among applications. The increasing popularity of XML Web services motivates us to examine if it is feasible to substitute one vendor service for another when using a Web-based application, assuming that these services are "derived from" a common base. If such substitution were possible, end users could use the same application with a variety of back-end vendor services, and the vendors themselves could compete on price, quality, availability, etc. And ontologies also can use in the autonomic agents and automatic reasoning

## 2.2 NER in Ontology Learning

Ontology learning is the automatic or semi-automatic process of extracting

ontology elements from large corpora of text. The elements are either presented as lists of candidates or mapped directly onto an appropriate ontology language. Ontology learning is an interdisciplinary task, typically, this task, starting from the terminology extraction, and usually includes several language processing (such as word segmentation, POS tagging, etc.); Then, through statistical[4] or rules to extract relations; the last the concepts and relations together constitute an ontology.

Named Entity Recognition can be used to automatically populate a legal ontology from legal texts following ontology learning [5]. Nicolas Weber[6] shows how web resources such as Wikipedia and Wiktionary can be used in combination with a domain corpus, a general purpose named-entity tagger and a seed or ‘base’ ontology to derive a domain ontology.

## 2.3 Alternatives to NER

To consider alternatives additionally to the gold standard, we can use combinations of Conditional Random Fields (CRF) together with a normalizing tagger.[7] A conditional random field (CRF) is a type of discriminative probabilistic model most often used for the labeling or parsing of sequential data, such as natural language text or biological sequences. Much like a Markov random field, a CRF is an undirected graphical model in which each vertex represents a random variable whose distribution is to be inferred, and each edge represents a dependency between two random variables. Alternative to NER by using CRF process is followed by a post processing step including an acronym disambiguation based on Latent Semantic Analysis (LSA). For robust model selection we apply 50-fold Bootstrapping to obtain an average F-Score of 84.58 % on the training set and 86.33 % on the test set.[7]

### 2.3.1 Information Extraction

Information Extraction is one kind of Information Retrieval that the target is to automatically extract structured information from unstructured machine readable documents, generally human language texts by means of natural language processing (NLP). Nowadays, IE focus on MUC conference. Less linguistically intensive approaches have been developed for IE on the Web using wrappers, which are sets of highly accurate rules that extract a particular page's content. There are several typical subtasks:

- **Named Entity Recognition**

Recognition of NE, this thesis will focus on this part.

- **Coreference resolution**

Detect of coreference and anaphoric links between text entities. It will find a typical link between previously extracted named entities. Such as “Norges teknisk-naturvitenskapelige universitet” and “NTNU” will consider as the same entities.

- **Terminology extraction**

Find a relevant term for a given corpus.

- **Relationship extraction**

Recognize the relations between entities.

## 2.4 Learning method of NER



The ability to identify previously unknown persons, is an essential component of NER systems. This ability depends on whether the detection and classification rules triggered by Features with positive and negative examples assigned. While early studies were mostly on craft rules that use the most recent monitoring machine learning as a way to induce automatic systems or rule-based sequence labeling algorithms based on a collection of examples of training. This is reflected in the scientific community, by the fact that five of the eight rule-based systems in the MUC-7 competition have been for sixteen systems were presented CONLL-2003, a forum dedicated to learning. When training samples are not available, hand-crafted rules remain the preferred technique, as shown in S. Sekine and Nobata (2004)[23], a system for 200 NER entity developed.

There are three main method of learning NE: Supervised Learning (SL), semi-supervised learning (SSL) and unsupervised learning (UL). The main shortcoming of SL is the requirement of a large annotated corpus. The unavailability of such resources and the prohibitive cost of creating them lead to two other alternative learning methods.

### 2.4.1 Supervised Learning

The idea of supervised learning is to study the features of positive and negative examples of NE over a large collection of annotated documents and design rules that capture instances of a given type. The current dominant technique for addressing the NER problem is supervised learning. SL techniques include Hidden Markov Models (HMM) [24], Decision Trees [25], Maximum Entropy Models (ME) [26], Support Vector Machines (SVM) [27], and Conditional Random Fields (CRF) [28]. These are all variants of the SL approach that typically consist of a system that reads a large annotated corpus, memorizes lists

of entities, and creates disambiguation rules based on discriminative features.

A baseline SL method, which is often proposed, consists of tagging words of a test corpus, if they are annotated as entities in the training data. The performance of the system depends on the baseline to be transferred to the vocabulary, with the percentage of words that appear without repetition, both in training and test corpus. D. Palmer and Day (1997) [29] calculates the vocabulary transfer to the MUC-6 training data. They report on a transfer of 21%, with as much as 42% of place names not repeated, but only 17% of the organizations and 13% of those names. Vocabulary transfer is a good indicator of the recall (number of people over the total number of units) identifies the baseline system, but is a pessimistic measure, because some bodies are often repeated in the documents. A. Mikheev et al. (1999)[30] is just the recall of the baseline system on the MUC-7 Corpus calculated. They report a recall of 76% for sites, 49% of organizations and 26% for people with precision of 70% to 90%. Whitelaw and Patrick (2003)[31] report consistent results on MUC-7 for the aggregated enamex class. For the three species together, the accuracy of precision 76% and the recall is 48%.

## 2.4.2 Semi-supervised Learning

The term "semi-supervision '(or' weak supervision") is still relatively young. The main SSL technology is called "bootstrapping" and includes a small measure of control, like a row of seeds, for the beginning of the learning process. For example, a system aimed at "disease names" could prompt the user to give a small number of example names. Then the system looks for sentences that contain these names, and tries to identify some clues from the context of five common examples. Then the system tries to other cases of the disease names that appear to be found in similar contexts. The learning curve is then reapplied to the

newly found examples, you discover relevant new contexts. By repeating this process, a large number of disease names and a variety of contexts will eventually be obtained. Recent experiments in semi-supervised NER [32] report that rival performances Baseline monitoring approaches. Here are some examples of SSL approaches.

S. Brin (1998) [33] by implementing lexical properties of regular expressions, paired to generate lists of book titles, book authors. It begins with examples such as seed (Isaac Asimov, The Robots of Dawn) and use some fixed lexical rules control how the following regular expression  $[A-Z][A-Za-z .,&] [A-Za-z.]$  used to describe a title. The basic idea of his algorithm is that many Web pages that correspond to a reasonable standard format via the website. If a particular site is found, the seed samples, which may contain new couples often face identified with simple constraints such as the presence of identical text, between or after the elements of an interesting pair. For example, the passage "The Robots of Dawn by Isaac Asimov (Paperback)" would allow, on the same site, "The Ants by Bernard Werber (Paperback)".

M. Collins and Singer (1999)[34] analyzes an entire corpus in search of candidates NE patterns. A pattern is, for example, a proper name (as determined part-of-speech tagger) by a noun phrase in apposition (eg, Maury Cooper, vice president at S & P), followed. Patterns are in pairs (spelling, where the context) denotes spelling of proper names, and here refers to the noun phrase kept in its context. Starting with an initial seed of the spelling (eg, Rule 1: If the spelling is "New York" then it is a place, and Rule 2: includes where the spell checker "Lord" then there is one person, and Rule 3: If the spelling is all then there is an organization), the candidates are tested on. Candidates meet a spelling rule, are classified according to their contexts and accumulated. The most common contexts are found shot in a series of contextual rules. Following the steps above,

contextual rules can be used to find other spelling rules, and so on. M. Collins and Singer[34], R. Yangarber et al. (2002)[35], show the idea that learning different types of NE while also allowing the identification of negative evidence (a kind of against all) and reduce over-generation. S. Cucerzan and Yarowsky (1999)[36], a similar technique and applies it to many languages.

E. Riloff and Jones (1999)[37] presented that the mutual bootstrapping a growing number of organizations and a number of contexts is again. Instead of using pre-defined candidate NE's (found through a fixed syntactic construct), they start with a handful of seeds unit examples of a particular type (eg Bolivia, Guatemala, Honduras are entities of type country) and all samples are enriched found around the seeds in a large corpus. Contexts (eg offices in X, X, in equipment ...) are arranged and used to find new examples. Riloff and Jones note that the performance of the algorithm to deteriorate rapidly when disturbances in the Entity List, or pattern-list introduced. While they report relatively low precision and recall in their experiments, the work proved to be very influential.

Pasca M. et al. (2006)[38] are also using techniques inspired by mutual bootstrapping. But they generate through the use of D. Lin 's innovations (1998) distributional effects similar synonyms - or, more generally, words that are members of the same semantic class - so that patterns generalization. For example, for the pattern X was born in November, Lin's synonyms for November (March, October, April, March, August, February, July, Nov., ...) so that the training of new patterns such as X was born in March. One of the contribution of Pasca et al[38]. is to apply the technique to very large corpora (100 million Web documents) and demonstrate that, from a starting capital of 10 examples of facts (as entities of type person with combined units of the type defined years - is for the person, the year of birth), it is a million facts with an accuracy of about 88% to generate.

The problem of selection of the unlabeled data is addressed by J. Heng and Grishman (2006)[39]. They show how an existing NE classifier may be using bootstrapping methods. The most important lesson they report is that trust in a large collection of documents is not sufficient. Selection of documents with information retrieval relevance-like measures and the selection of specific contexts, the rich get to be proper names and Coreference the best results in their experiments.

### 2.4.3 Unsupervised Learning

The typical approach to unsupervised learning is clustering. For example, one can try to collect names from clustered groups based on the similarity of context. There are other methods also unattended. Basically, the techniques based on lexical resources (eg WordNet), calculated on lexical patterns and statistics on a large unannotated corpus. Here are some examples.

E. Alfonseca and Manandhar study (2002)[40], the problem of labeling an input with a corresponding word NE type. NE-types from WordNet (eg taken place> Land, animate "person, animate> Animals, etc.). The approach is to assign a theme to each WordNet synset signature by simply listing words that occur frequently together with him in a large corpus. Then, as a command word will appear in a given document, the word context (words in a fixed-size window around the input word) to the type signature is compared and classified among the similar.

Y. Shinyama and Sekine (2004)[41] uses an observation that these bodies often appear simultaneously in several news articles, while not common nouns. You

found a strong correlation between a name and unit on time (in time) and simultaneously in multiple news sources. This technique permits the identification of rare proper names in an unsupervised way, and in combination with other useful NER methods.

# Chapter 3

## Named Entity Recognition

At first, Named Entity Recognition (NER) was present as a subtask of MUC-6(Message Understanding Conference). NER is also known as entity identification and entity extraction. The task of NER is identifying and classifying the proper names appearing in the text and the number of meaningful phrases.

### 3.1 Named Entity Recognition Systems

The first paper of research NER was presented at the Seventh IEEE Conference on Artificial Intelligence Applications by Lisa F. Rau (1991)[42]. Rau's paper describe a system that "extract and recognize [company] names", it relies on heuristics and handcrafted rules. From 1996, with the first major in task MUC-6, it never declined since then with steady research and numerous scientific events: HUB-4, MUC-7 and MET-2, IREX, CONLL, ACE and HAREM. The Language Resources and Evaluation Conference (LREC) has also been staging workshops and main conference tracks on the topic since 2000.

Named Entity Recognition Systems have been created that use linguistic grammar-based techniques and statistical models. Handcrafted grammar-based systems are usually obtained better precision, however, lower recall in months of work by experienced linguists cost calculation. Statistical NER systems typically require a large amount of manually annotated training data. It usually find the sequence of tags that maximizes the probability  $p(N|S)$ , where  $S$  is the sequence of words in a sentence, and  $N$  is the sequence of named-entity tags assigned to the words in  $S$ . [8]

At first, English is the most popular language factor to research NER, but along with the development of research in these areas, more and more kinds of language have been researched. German is well studied in CONLL-2003 and in earlier works. Similarly, Spanish and Dutch are strongly represented, boosted by a major devoted conference: CONLL-2002. Japanese has been studied in the MUC-6 conference, the IREX conference and other work. Chinese is studied in an abundant literature [43], and so are French [44], Greek [45] and Italian [46]. And then many other language has paid more attention on this area. Finally, Arabic has started to receive a lot of attention in large-scale projects such as Global Autonomous Language Exploitation (GALE)

Parts of Speech taggers (POS taggers), also called word-category disambiguation. It reads text in some language and assigns parts of speech to each word (and other token), such as noun, verb, adjective, etc. It based on both its definition, as well as its context, for example, relationship with adjacent and related words in a phrase, sentence, or paragraph. Labeling part of speech is more difficult than simply having a list of words and their parts of speech, because some words can represent more than one part of speech at different times. For example, the word “work”, can be considered as noun or verb. Some of the NER systems are



incorporated into POS taggers. Moreover, most of the NER systems are based on analyzing patterns of POS taggers.

## 3.2 Named Entity types

In the expression “Named Entity”, the word “Named” aims to restrict the task to only those entities for which one or many rigid designators, as defined by S. Kripke (1982) [49], stands for the referent. For instance, the automotive company created by Henry Ford in 1903 is referred to as Ford or Ford Motor Company. Rigid proper names and certain identifiers are natural kind terms such as biological species and substances. There is a general agreement in the NER community on the inclusion of temporal expressions and some numerical expressions such as money and other types of units. While some instances of these types are good examples of rigid designators (e.g., the year 2010 is the 2010th year of the Gregorian calendar) there are also many invalid ones (e.g., in June refers to the month of an undefined year – past June, this June, June 2010, etc.). It is arguable that the NE definition is loosened in such cases for practical reasons.

Early work formulates the NER problem as recognizing “proper names” in general. Overall, the most studied types are three specializations of “proper names”: names of “persons”, “locations” and “organizations”. These types are collectively known as “enamel” since the MUC-6 competition. The type “location” can in turn be divided into multiple subtypes of “finegrained locations”: city, state, country, etc. [50]. Similarly, “fine-grained person” sub-categories like “politician” and “entertainer” appear in the work of M. Fleischman and Hovy (2002)[51]. The type “person” is quite common and used at least once in an original way by O. Bodenreider and Zweigenbaum (2000)[52] who combines it with other cues for extracting medication and disease names (e.g., “Parkinson disease”). In the ACE

program, the type “facility” subsumes entities of the types “location” and “organization”. The type “GPE” is used to represent a location which has a government, such as a city or a country.

The type “miscellaneous” is used in the CONLL conferences and includes proper names falling outside the classic “enamel”. The class is also sometimes augmented with the type “product”. The “timex” (another term coined in MUC) types “date” and “time” and the “numex” types “money” and “percent” are also quite predominant in the literature. Since 2003, a community named TIMEX2 [53] proposes an elaborated standard for the annotation and normalization of temporal expressions.

A recent interest in bioinformatics, and the availability of the GENIA corpus [54] led to many studies dedicated to types such as “protein”, “DNA”, “RNA”, “cell line” and “cell type” as well as studies targeted to “protein” recognition only [55]. Related work also includes “drug” [56] and “chemical” [57] names.

Some recent work does not limit the possible types to extract and is referred as “open domain” NER. In this line of research, S. Sekine and Nobata (2004)[23] defined a named entity hierarchy which includes many fine grained subcategories, such as museum, river or airport, and adds a wide range of categories, such as product and event, as well as substance, animal, religion or color. It tries to cover most frequent name types and rigid designators appearing in a newspaper. The number of categories is about 200, and they are now defining popular attributes for each category to make it an ontology.

The first Named Entity set had 7 types [9], organization, location, person, date, time, money and percent expressions. There is a general agreement to include temporal expressions and some numerical expressions (i.e., money, percentages, etc.) as instances of named entities in the context of the NER task.

The number of types of entities such was limited because the target application of the evaluation was to extract information for business activities. There were many sub-types for that category. We can consider the domain of these types as hierarchies of Named entity. Then, S. Sekine merged the hierarchies into numerical expressions and name type expressions.[10]

### 3.3 Applications of Named Entity Recognition

A NER is useful in many Natural Language Processing applications such as information extraction, question answering, parsing, machine translation, the metadata for the Semantic Web mark an important foundation. On its own, a NER can also provide users who are looking for person or organization names with quick information.[8] Usually, NER systems are used in the areas of entity identification in the molecular biology, bioinformatics, and medical natural language processing communities. Early time, NER systems were used by primarily extraction from journalistic articles, and then Automatic Content Extraction (ACE) evaluation also included several types of informal text styles, such as weblogs and text transcripts from conversational telephone speech conversations.

Using in the areas of textual genre (journalistic, scientific, informal, etc.) and domain (gardening, sports, business, etc.), has been rather neglected in the NER literature. Few studies are specifically devoted to diverse genres and domains. D. Maynard *et al.* (2001)[58] designed a system for emails, scientific texts and religious texts. E. Minkov *et al.* (2005)[59] created a system specifically designed for email documents. Perhaps unsurprisingly, these experiments demonstrated that although any domain can be reasonably supported, porting a system to a new

domain or textual genre remains a major challenge. T. Poibeau and Kosseim (2001)[60], for instance, tested some systems on both the MUC-6 collection composed of newswire texts, and on a proprietary corpus made of manual translations of phone conversations and technical emails. They report a drop in performance for every system (some 20% to 40% of precision and recall).

### 3.4 Feature space of NER

Features are characteristic attributes of words designed for algorithmic consumption. An example of a function is a Boolean variable with the value *true* if a word is activated, and *false* otherwise. Feature vector representation is an abstraction over the text, which usually represented each word by one or many Boolean, numerical and nominal values. For example, a hypothetical system NER represent each word of a text with three attributes [22]:

- a Boolean attribute with the value *true* if the word is capitalized and *false* otherwise;
- a numeric attribute corresponding to the length, in characters, of the word;
- a nominal attribute corresponding to the lowercased version of the word.

Normally, the NER has solved problems in the application of a rule-system of functions. For example, a system has two rules, a recognition rule: enabled, are words candidate organizations "and a classification rule," the kind of candidate units of length more than three words organization. "Those rules are good for the prototype set before. However, real systems tend to be much more complex and their rules are often created by automated learning. Usually, there are three different features to recognize NE: Word-level features, List lookup features and Document and corpus features.

### 3.4.1 Word-level features

Word-level features are related to the character makeup of words. They specifically describe word case, punctuation, numerical value and special characters. It contains several features below.

- Digit Pattern

Digits can express a wide range of useful information such as dates, percentages, intervals, identifiers, etc.

- Common word ending

Morphological features are essentially related to words affixes and roots. For instance, a system may learn that a human profession often ends in “ist” (journalist, cyclist) or that nationality and languages often ends in “ish” and “an” (Spanish, Danish, Romanian).

- Functions over word

Features can be extracted by applying functions over words

- Patterns and summarized patterns

The role of Pattern features is to map words onto a small set of patterns over character types.

### 3.4.2 List Look up Features

Lists are the privileged features in NER. The terms “gazetteer”, “lexicon” and “dictionary” are often used interchangeably with the term “list”. List inclusion is a

way to express the relation “is a” (e.g., Trondheim is a city). It may appear obvious that if a word (Trondheim) is an element of a list of cities, then the probability of this word to be city, in a given text, is high. However, because of word polysemy, the probability is almost never (e.g., the probability of “Fast” to represent a company is low because of the common adjective “fast” that is much more frequent).

We could enumerate many more list examples but we decided to concentrate on those aimed at recognizing enamex types.

- General Dictionary

Common nouns listed in a dictionary are useful, for instance, in the disambiguation of capitalized words in ambiguous positions

- Words that are typical of organization names

Many authors propose to recognize organizations by identifying words that are frequently used in their names.

- On the list lookup techniques

Most approaches implicitly require candidate words to exactly match at least one element of a pre-existing list. However, we may want to allow some flexibility in the match conditions. At least three alternate lookup strategies are used in the NER field: word can be stemmed, “fuzzy-matched” and accessed using the Soundex algorithm.

### 3.4.3 Document and corpus features

Document features are defined over both document content and document structure.

Large collections of documents (corpora) are also excellent sources of features. We list in this section features that go beyond the single word and multi-word expression and include meta-information about documents and corpus statistics.

- Multiple occurrences and multiple casing
- Entity coreference and alias
- Document meta-information
- Statistics for Multiword units





# Chapter 4

## Existing Systems

This Chapter will introduce the definition and character of existing systems for Named Entity Recognition. Among the internet, there are three chosen systems for this project to evaluate: GATE, LbjNerTagger and CRFClassifier.

The table below shows the main introduction and comparison of these three systems.

	Quality	Flexibility	Set of types supported	Integration with other components	Languages supported	Screenshots from NER
GATE	See below	As part of the function of GATE, NEs can be recognized by ANNIE(Named-Entity State	Person, location, organization, Ambiguities, date,	It is compactness with other components cause	English	See below

		Machine Patterns ); use JAPE language	number, address, url, identifier, jobtitle	of thorough system		
LbjNerTag ger	See below	Can easily load a file to analysis and output the NEs recognize file.	Person, Location, Organizati on, date, number	Using java script, close to other compone nts	English	See below
CRFClassi fier	See below	Forthright, easy to understand and use	Person, Location, Organizati on, MISC	Very easy interface, only show and analysis files for NER	English	See below

Table 4.1 Character of each systems

## 4.1 GATE

Screenshoot:

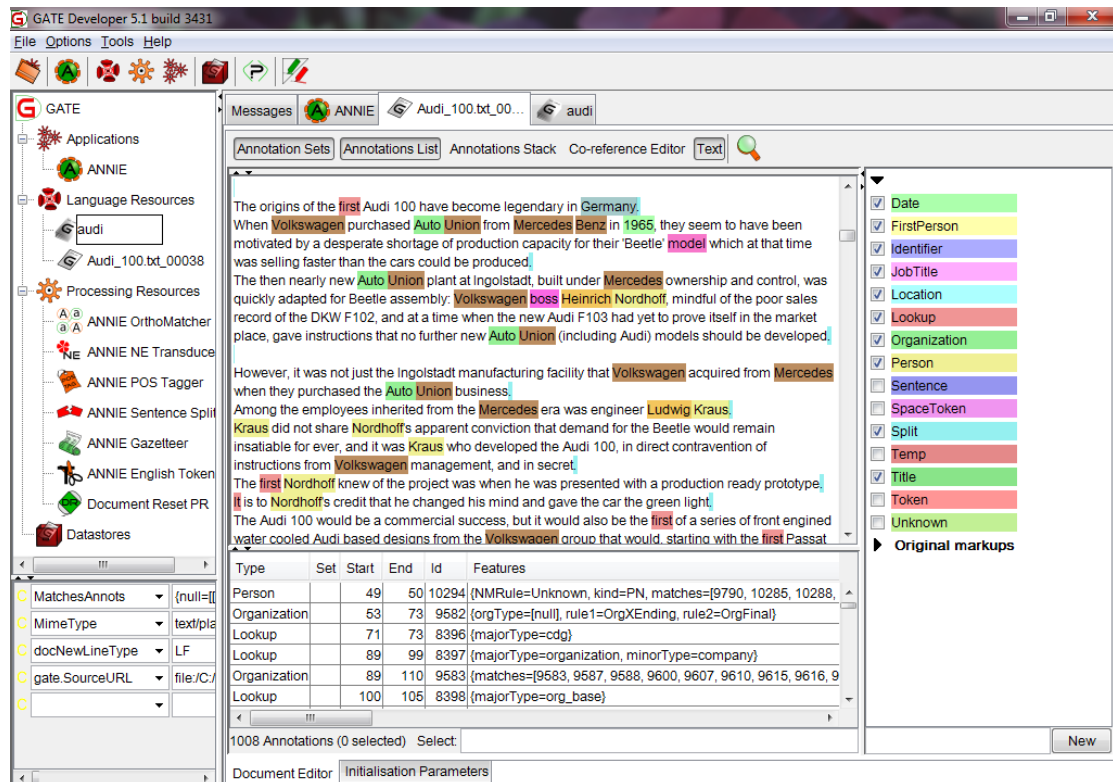


Figure 4.1 Screenshot of GATE

General Architecture for Text Engineering (GATE), developed by The University of Sheffield, is a framework for text analysis developed in JAVA, available as open-source software. GATE is an infrastructure for developing and deploying software components that process human language.[11] GATE is an infrastructure for developing and deploying software components that process human language. It is nearly 15 years old and is in active use for all types of computational task involving human language. GATE excels at text analysis of all shapes and sizes. From large corporations to small startups, from €multi-million research consortia to undergraduate projects, our user community is the largest and most diverse of any system of this type, and is spread across all but one of the continents.

GATE is open source free software; users can obtain free support from the user and developer community via GATE.ac.uk or on a commercial basis from our industrial partners.

GATE is not only a framework for text engineering, but also is an architecture and a development environment. ANNIE, a Nearly-New Information Extraction System, that is distributed with an IE system by GATE. NER is one of function in ANNIE. These recourses can be used as one unit or used as individual components along with others. ANNIE consists of the following processing components for English text:

- Tokenizer

The tokenizer splits the text into very simple tokens such as numbers, punctuation and words of different types.

In the default set of rules, the following kinds of Token and SpaceToken are possible: word; number; symbol; Punctuation; SpaceToken. Also, there's an English Tokeniser in this system. It is a processing resource that comprises a normal tokeniser and a JAPE transducer. The transducer has the role of adapting the generic output of the tokeniser to the requirements of the English part-of-speech tagger.

- Sentence splitter

The sentence splitter, which is domain and application-independent, is a cascade of finite state transducers which segments the text into sentences. This module is required for the tagger. The splitter uses a gazetteer list of abbreviations to help distinguish phrase marked points and apart from other types

- Part-of Speech tagger

POS tagger was introduced before that used to recognize parts of speech to each word. Regarding to ANNIE of GATE, it produces a part-of-speech tag as an annotation on each word or symbol. The tagger uses a default lexicon and ruleset (the result of training on a large corpus taken from the Wall Street Journal).

- Gazetteer

The gazetteer lists used are plain text files, with one entry per line. Each list represents a set of names, such as names of cities, organisations, days of the week, etc.

The ANNIE gazetteer is part of and provided by the ANNIE plugin. Each individual gazetteer list is a plain text file, with one entry per line. Below is a section of the list for units of currency:

Ecu

European Currency Units

FFr

Fr

German mark

German marks

New Taiwan dollar

New Taiwan dollars

NT dollar

NT dollars

- Semantic tagger

ANNIE's semantic tagger is based on the JAPE language(JAPE is a Java Annotation Patterns Engine. JAPE provides finite state transduction over annotations based on regular expressions.). It contains rules which act on annotations assigned in earlier phases, in order to produce outputs of annotated entities.

- Orthomatcher

Orthomatcher named NameMatcher before. Its module adds identity relations between named entities found by the semantic tagger, in order to perform

coreference. It does not find new named entities as such, but it may assign a type to an unclassified proper name, using the type of a matching name.

### ● Coreferencer

The pronominal coreference module performs anaphora resolution using the JAPE grammar formalism.

The main coreference module can operate successfully only if all ANNIE modules were already executed. The module depends on the following annotations created from the respective ANNIE modules:

- ✧ Token (English Tokenizer)
- ✧ Sentence (Sentence Splitter)
- ✧ Split (Sentence Splitter)
- ✧ Location (NE Transducer, OrthoMatcher)
- ✧ Person (NE Transducer, OrthoMatcher)
- ✧ Organization (NE Transducer, OrthoMatcher)

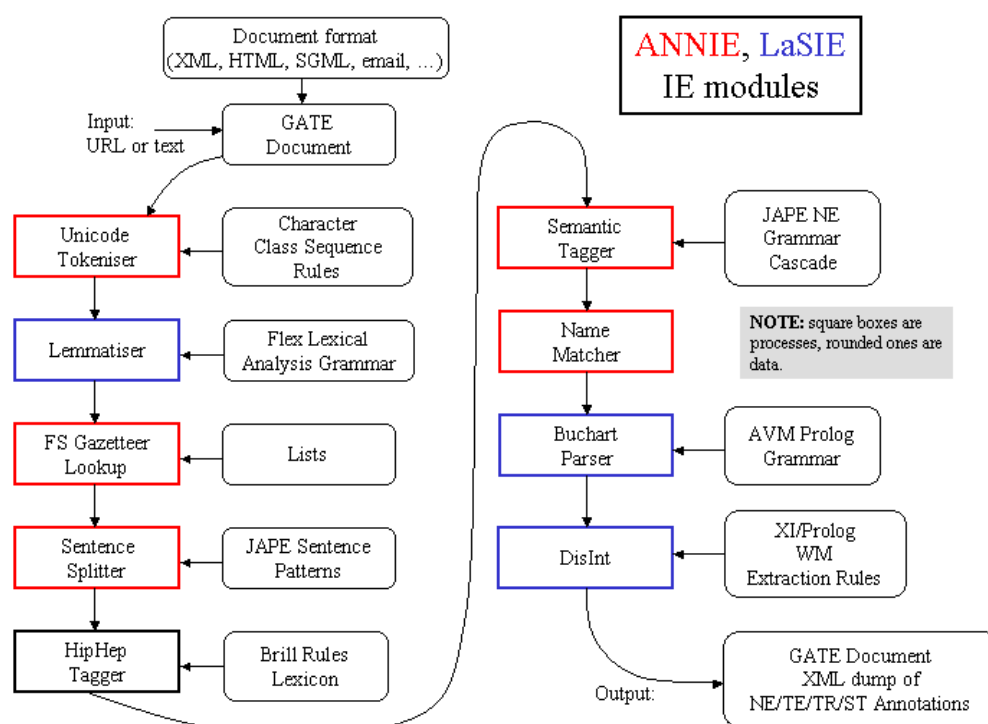


Figure 4.2 The pipeline of ANNIE components

ANNIE relies on finite state algorithms and the JAPE language. A JAPE grammar consists of a set of phases, each of which consists of a set of pattern/action rules. The phases run sequentially and constitute a cascade of finite state transducers over annotations. The left-hand-side (LHS) of the rules consist of an annotation pattern description. The right-hand-side (RHS) consists of annotation manipulation statements. Annotations matched on the LHS of a rule may be referred to on the RHS by means of labels that are attached to pattern elements.

A document or a corpus can be annotated and stored when running these components. Otherwise, GATE comes with a large set of plug-ins that can be loaded at any time. These include:

- Ontology Editor
- Machine Learning component
- WordNet component
- Information Retrieval component
- Stemmer with support for several languages
- Noun Phrase Chunker
- TreeTagger, another Part-of-Speech tagger with support for several languages.

Quality:

D. Maynard described an experiment to adapt a NER system from English to Cebuano as part of the TIDES surprise language program. They use ANNIE system for Cebuano and achieved an F-measure of 77.5%. [15] K. Bontcheva present the shallow methods for named entity coreference, which we developed as modules in the ANNIE Information Extraction system. [16] D. Maynard also presented the GATE architecture and framework for Language Engineering, and the MUSE cross-genre Information Extraction system developed within GATE. [17] K. Pastra

discussed the feasibility of reusing grammars for Named Entity Recognition by GATE.[18]

Tag Sets: <person>person</person>; <location>location</location>, etc. The tags will relate with each other by the JAPE language.

## 4.2 LbjNerTagger[12]

Screenshot:

```
Platform
[ORG Volkswagen Group ] C1 platform
[ORG Audi ] 100 [ORG LS ] 2-door
The origins of the first [ORG Audi ] 100 have become legendary in [LOC
Germany ] .
When [ORG Volkswagen ] purchased [ORG Auto Union ] from [ORG Mercedes
Benz ] in 1965 , they seem to have been motivated by a desperate shortage
of production capacity for their 'Beetle' model which at that time was
selling faster than the cars could be produced .
The then nearly new [ORG Auto Union ] plant at [LOC Ingolstadt ] , built
under [MISC Mercedes ] ownership and control , was quickly adapted for
[ORG Beetle ] assembly: [ORG Volkswagen ] boss [PER Heinrich Nordhoff ]
, mindful of the poor sales record of the [MISC DKW F102 ] , and at a time
when the new [ORG Audi ] F103 had yet to prove itself in the market place
, gave instructions that no further new [ORG Auto Union ] ( including [ORG
Audi ] ) models should be developed .
However , it was not just the [LOC Ingolstadt ] manufacturing facility
that [ORG Volkswagen ] acquired from [MISC Mercedes ] when they purchased
the [ORG Auto Union ] business .
Among the employees inherited from the [MISC Mercedes ] era was engineer
[PER Ludwig Kraus ] .
[PER Kraus ] did not share [PER Nordhoff ] 's apparent conviction that
demand for the [ORG Beetle ] would remain insatiable for ever , and it was
[PER Kraus ] who developed the [ORG Audi ] 100 , in direct contravention
of instructions from [ORG Volkswagen ] management , and in secret .
The first [PER Nordhoff ] knew of the project was when he was presented
with a production ready prototype .
```

Figure 4.3 Screenshot of LbjNerTagger

Illinois Named Entity Tagger(LBJ based), which developed by University of Illinois at Urbana-Champaign, is a state of the art NER tagger that tags plain text with named entitites (people / organizations / locations / miscellaneous). It uses gazetteers extracted from Wikipedia, word class model derived from unlabeled text and expressive non-local features. The best performance is 90.8  $F_1$  on the



CoNLL03 shared task data. The tagger is robust and has been evaluated on a variety of datasets.

Learning Based Java(LBJ)[13] is a modeling language for the rapid development of software systems with one or more learned functions, designed for use with the Java<sup>TM</sup> programming language. LBJ offers a convenient, declarative syntax for classifier and constraint definition directly in terms of the objects in the programmer's application. With LBJ, the details of feature extraction, learning, model evaluation, and inference are all abstracted away from the programmer, leaving him to reason more directly about his application.

A classifier may be defined by:

- coding it explicitly in Java,
- using operators to build it from existing classifiers, or
- identifying feature extraction classifiers and a data source to learn it over.

Under the LBJ programming philosophy, the designer of a learning based program will first design an object oriented internal representation (IR) of the application's raw data using pure Java. A classifier is then any method that produces one or more discrete or real valued classifications with respect to a single object from the programmer's IR. Using LBJ, these classifications are easily interpretable either at face value as the application requires or as features amenable for input to a learning algorithm. Learning algorithms are employed to create learning classifiers, which are classifiers that can change their representation with experience. Once the LBJ compiler has generated these representations from their specifications and user supplied training objects, the application, written in pure Java, simply invokes any classifier on an IR object just like any other method. Programming with LBJ, the practitioner reasons in terms of his data directly, disregarding the cumbersome implementation details of feature extraction and learning.

This release allows us to annotate data with four flavors of pre-compiled models and to train an NER tagger with 4 different configurations:

- Config/baselineFeatures.config.
- Config/allLevel1.config
- Config/allFeatures.config
- Config/allFeaturesBigTrain

The baseline model achieves modest 83.6 F1 score on CoNLL03 test set. The "allLevel1" model is a one-layer model, which achieves 90.25F score on CoNLL03 shared task. The "allFeatures" model is a two-layer architecture that is considerably slower, and marginally better, achieving 90.5 F1 score on the CoNLL03 shared task. The last model is also a two-layer model, it uses the same features as the previous one, but it was trained both on training and the development set of the CoNLL03 dataset. It achieves 90.8F1 score on the CoNLL03 test set.

Quality:

N/A

Tag Sets: [PER person]; [LOC location]; [ORG organization]. The different tags relate with each other by Learning Based Java program.

## 4.3 CRFClassifier[14]

Screenshot:

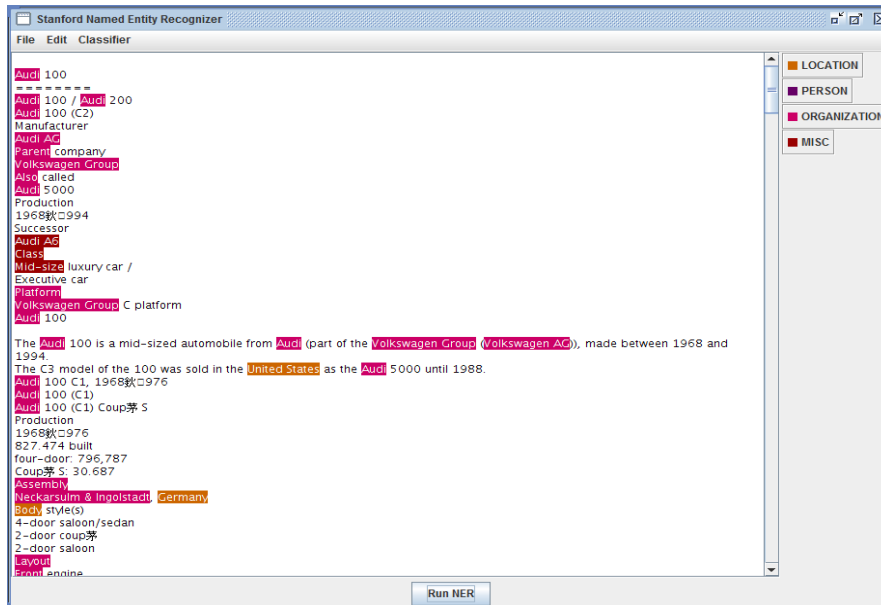


Figure 4.4 Screenshot of CRFClassifier

CRFClassifier is a Java implementation of a Named Entity Recognizer. It is developed by Jenny Finkel in University of Stanford. The feature extractors are by Dan Klein, Christopher Manning, and Jenny Finkel. Much of the documentation and usability is due to Anna Rafferty. The software provides a general (arbitrary order) implementation of linear chain Conditional Random Field (CRF) sequence models, coupled with well-engineered feature extractors for Named Entity Recognition. The software provided here is similar to the baseline local+Viterbi model in that paper, but adds new distributional similarity based features (in the -distSim classifiers). The big models were trained on a mixture of CoNLL, MUC-6, MUC-7 and ACE named entity corpora, and as a result the models are fairly robust across domains.

At the beginning, it worked on a wide range of NER and IE related tasks over the past several years. The University of Stanford entered the 2003 CoNLL NER shared task, using a Character-based Maximum Entropy Markov Model (MEMM). In late 2003 we entered the BioCreative shared task, which aimed at doing NER in the domain of Biomedical papers. This task required identifying genes and proteins, but not distinguishing between the two. We used a similar model as for the CoNLL shared

task, but more tuned to the domain and with some additional features; they had the best performing system. Then, in 2004, they entered the BioNLP shared task at CoLing which also looked at Biomedical papers, but required identifying five different classes - DNA, RNA, cell line, cell type, and protein. They once again used an MEMM, but added much richer features, including features from parse trees, the web, and how entities were labeled elsewhere on a previous run. They also entered the PASCAL IE shared task, which involved extracting information from workshop announcements. They attempted to use a relational model in addition to the MEMM to allow the use of top-down information. They have also studied the use of Gibbs sampling for inference in a Conditional Random Field (CRF), so as to incorporate longer distance information. There has also been work on adapting sequence classifiers to new, unseen domains.

The basic CRF model follows that of Lafferty et al.(2001). The reason they choose a CRF because it represents the state of the art in sequence modeling, allowing both discriminative training and the bi-directional flow of probabilistic information across the sequence. A CRF is a conditional sequence model which represents the probability of a hidden state sequence given some observations. In order to facilitate obtaining the conditional probabilities they need for Gibbs sampling, they generalize the CRF model in a way that is consistent with the Markov Network literature (see Cowell et al. (1999)): they create a linear chain of cliques, where each clique,  $c$ , represents the probabilistic relationship between an adjacent pair of states<sup>2</sup> using a clique potential  $\psi_c$ , which is just a table containing a value for each possible state assignment.

The table is not a true probability distribution, as it only accounts for local interactions within the clique. The clique potentials themselves are defined in terms of exponential models conditioned on features of the observation sequence, and must be instantiated for each new observation sequence. The sequence of potentials in the clique chain then defines the probability of a state sequence (given the observation

sequence) as:

$$P_{\text{CRF}}(\mathbf{s}|\mathbf{o}) \propto \prod_{i=1}^N \phi_i(s_{i-1}, s_i)$$

where  $\phi_i(s_{i-1}, s_i)$  is the element of the clique potential at position  $i$  corresponding to states  $s_{i-1}$  and  $s_i$ .

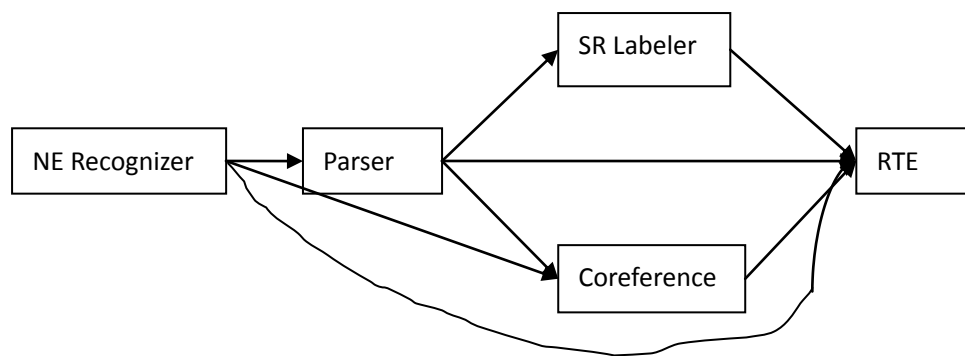


Figure 4.5 Textual Entailment Pipeline(Topological sort of annotators)

	Speed	Discrim vs. Generative	Normalization
HMM	Very fast	Generative	local
MEMM	Mid range	Discriminative	local
CRF	Kinda slow	Discriminative	globe

Table 4.2 Model Trades-off

CRFClassifier use current word, previous word, next word, all words within a window as word features, use two form as orthographic features. For example: Jenny (Xxxx) and IL-2 (XX-#). It also has prefixes and suffixes, such as Jenny <J, <Je,

<Jen, ..., nny>, ny>, y>. Label sequences and Lots of feature conjunctions can also find by using in CRFClassifier.

Feature	NER
Current Word	Yes
Previous Word	Yes
Next Word	Yes
Current Word Character n-gram	all
Current POS Tag	Yes
Surrounding POS Tag Sequence	Yes
Current Word Shape	Yes
Surrounding Word Shape Sequence	Yes
Presence of Word in Left Window	Size 4
Presence of Word in Right Window	Size 4

Table 4.3 Features used by the CRF for named entity recognition (NER)

Quality:

Shipra Dingare and Jenny Finkel present the way of using CRFClassifier to identifying NER in biomedical text.[19][20][21]

Tag Set: Same as GATE, <person>person</person>, etc. They relate with each other by CRF model.

# Chapter 5

## Evaluation of Existing Systems

This chapter is described how to evaluate these three existing systems by developed program. The goal of this evaluation is trying to find the advantage and disadvantage of these three systems, and then figure out which entity types that can be supported by the different systems. We also want to investigate to what extent lists of typed entities and regular expressions may be used by the systems.

### 5.1 Corpus for evaluation

A suitable corpus should be collected to evaluate the existing systems to compare with each other. The suitable corpus must have the document that easy to understand and classical about the type of named entities. Every document of corpus must have the relationship with each other.

#### 5.1.1 Collection of corpus

After discussion of reasonable situation, “Audi” corpus is collected for this thesis. It has been selected in the Wikimedia website. According to the types of cars, it

lists in the different documents. Because of the evaluation needed, it should be merge into one document as a corpus.

### 5.1.2 Characterization of corpus

There are two sets of documents, one containing the documents tagged with 'Audi'. The tagged files are tagged by "Tagged and Cleaned Wikipedia". The zip-file contains the following structure:

- + cars
  - + tagged
  - + plain

In the plain files, the corpus have skipped all text that are in "infoboxes" in Wikipedia, as well as all the "External links" text. All files are named with the "html-name" given in the corpus; the tagged files have ".html" extension, the plain files have .txt extension. The number of documents: 90. The number of tags: 90. We use the files under the "plain" folder to evaluate the systems. Because the documents are huge, we need to merge them into one file. The size of final file is 609 KB.

## 5.2 Evaluation Method

The evaluation method should be processed by a developed program that calculate value of Recall and Precision so that to find the which named entities that can be supported by the different systems and what extent lists of typed entities and regular expressions may be used by the systems. We can simplify find the flow in Figure 3.



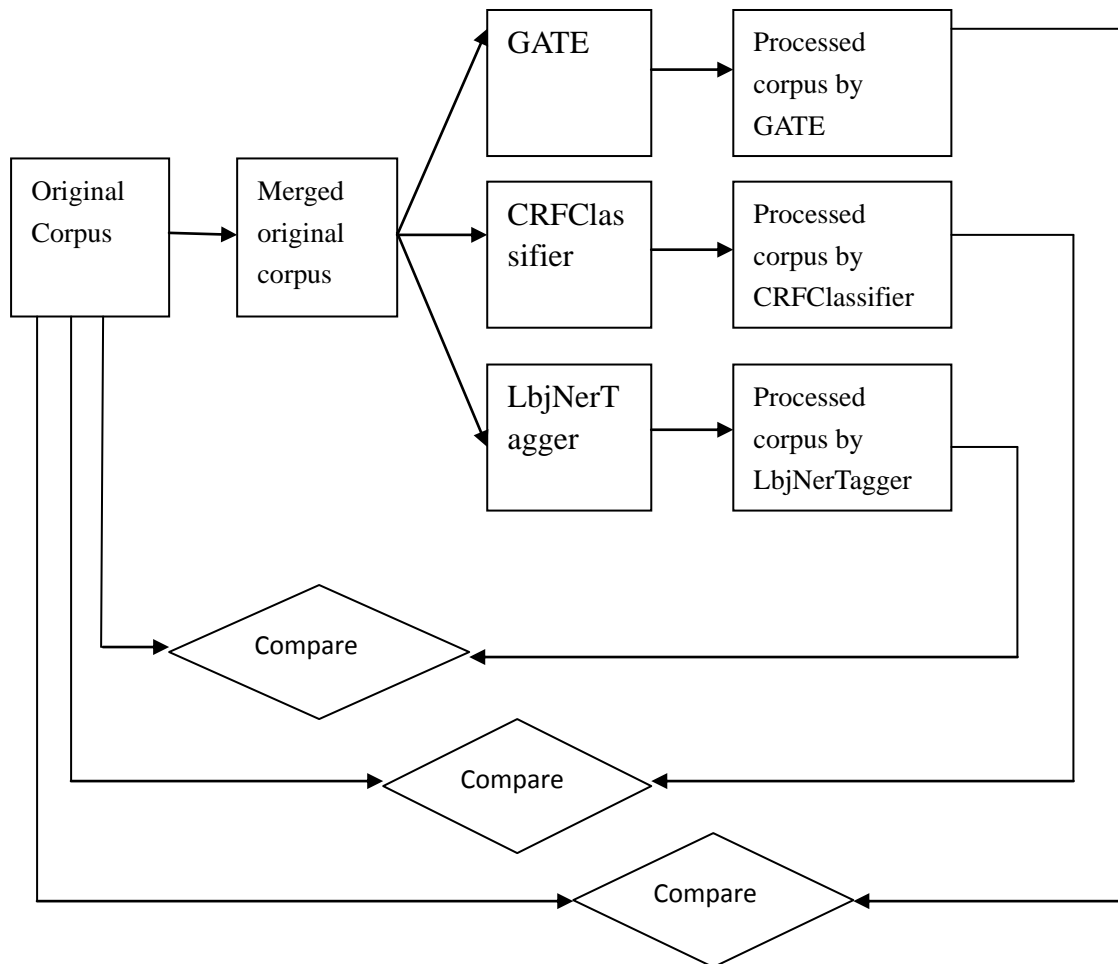


Figure 5.1 Process of evaluation

The “compare” process by calculate the value of recall and precision. Precision and recall are two widely used statistical classifications. In an information retrieval scenario, Precision is defined as the number of relevant documents retrieved by a search divided by the total number of documents retrieved by that search, and Recall is defined as the number of relevant documents retrieved by a search divided by the total number of existing relevant documents (which should have been retrieved).

In Information Retrieval contexts, Precision and Recall are defined in terms of a set of retrieved documents (e.g. the list of documents produced by a web search engine for a query) and a set of relevant documents (e.g. the list of all documents on the internet that are relevant for a certain topic).

$$\text{recall} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{relevant documents}\}|}$$

$$\text{precision} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{retrieved documents}\}|}$$

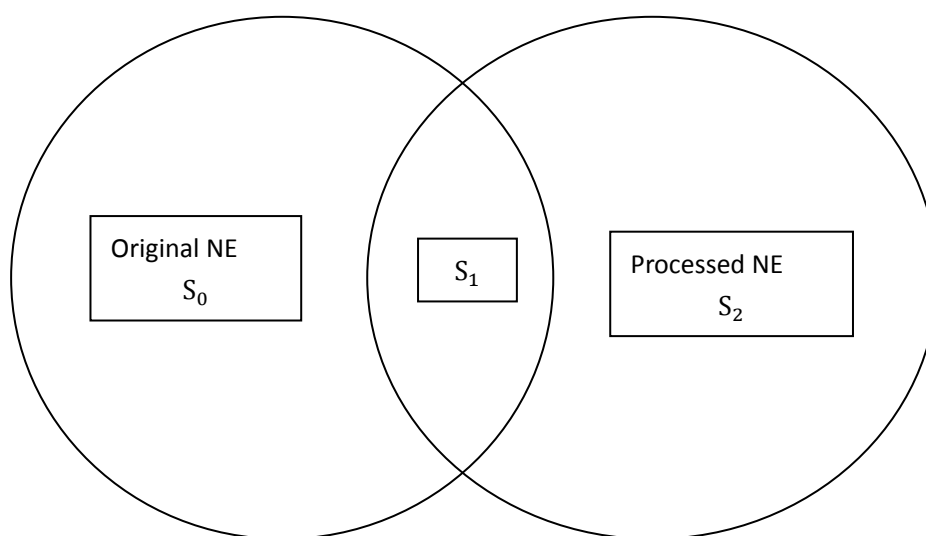


Figure 5.2 Definition of Recall and Precision

In terms of Figure 4, Recall and Precision can be defined as:

$$\text{Precision} = \frac{S_1}{S_2}; \text{ Recall} = \frac{S_0}{S_2}$$

We need to find the value of  $S_0$ ,  $S_1$  and  $S_2$  for Named entities. According this case, we have to develop a program to find.

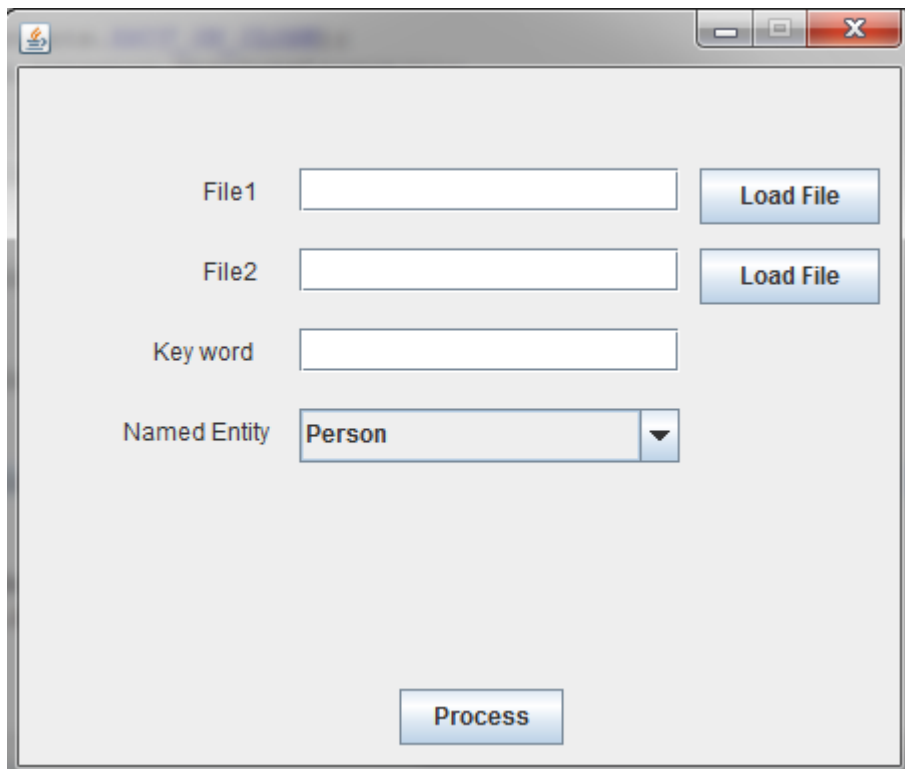
This program is developed by JAVA. The interface (Figure 5) of this program has three parts. First, the file load part. Two files should be loaded in this part, one is the original corpus, and another is processed corpus by one NER systems.

Second, is the “Key word” part. This part should input a keyword that it is a named entity.

Last, is the type of NE part. This part will choose which type I will process. According to the existing systems, I only define three type: Person, Location and Organization.

There will output three value, one (named a) is the number of key word founded in the original corpus; and b is quantity of key word founded that has been signed as a named entity in File2, which is the processed by existing systems; last one (named c) is the quantity of key word founded that has been signed as the right named entity which the user choose in the input interface in File2.

Because I want to find the value of  $S_0$ ,  $S_1$  and  $S_2$ . We can find that  $S_0$  is the value a.  $S_1$  is the value b.  $S_2$  is the value c.



The screenshot shows a Java application window with a light gray background. It contains four input fields and three buttons. The first two rows are for file loading: 'File1' and 'File2' each have a text input field followed by a 'Load File' button. The third row is for the keyword: 'Key word' has a text input field. The fourth row is for the named entity type: 'Named Entity' has a dropdown menu currently showing 'Person'. At the bottom center is a 'Process' button. The window has standard Windows-style title bar controls (minimize, maximize, close) in the top right corner.

Figure 5.3 Interface of developed program

I will present it as the table below for each existing system.

	Recall	Precision
Person		
Location		
Organization		

Table 5.1 Evaluation table

# Chapter 6

## Evaluation Results

This chapter will present and discuss the evaluation result among the developed program.

### 6.1 Concepts

Key word should be defined as a simple word that recognized easily through types of NE. we only evaluate the iconic Named entities for the existing systems, it is better to evaluate the all NEs. But it is hard to find the right way to definition the whole right Named entities in a huge corpus, cause only the artificial definition should be acceptant to as the standard NEs compare with the NEs that recognized by existing system. CRFClassifier is generally accepted system in the world now. a better system should be developed in the future. I defined a list of key words below (Named entities) to evaluate the existing systems.

Organization	Location	Person
Audi	Germany	Johann

Volkswagen Group	Hungary	Franz
NSU	Belgium	Caddy
Auto Union	China	Jetta
Volkswagen	India	Laurent
DTM	Japan	Rasmussen
Honda	United Kingdom	Felix Wankel
BMW	Asia	Heinrich Nordhoff
Toyota	Zwickau	Ludwig Kraus
Ford	USA	Eberhard Kittler

Table 6.1 The key words should be test

These key words can be accepted as their types extensively. We can use these key words easily to evaluate the systems by developed program.

## 6.2 Evaluation results of GATE

We can find the results through Table below.

Organization	Recall	Precision
Audi	100%	56%
Volkswagen Group	100%	60%
NSU	100%	1.8%
Auto Union	100%	86%
Volkswagen	100%	37%
DTM	0	0
Honda	100%	100%
BMW	100%	96%
Toyota	100%	94%

Ford	100%	71%
------	------	-----

Table 6.2 Evaluation results of Organization for GATE

Location	Recall	Precision
Germany	100%	86%
Hungary	100%	78%
Belgium	100%	100%
China	100%	70%
India	100%	75%
Japan	100%	44%
United Kingdom	100%	100%
Asia	100%	50%
Zwickau	100%	2.1%
USA	100%	67%

Table 6.3 Evaluation results of Location for GATE

Person	Recall	Precision
Johann	20%	100%
Franz	20%	100%
Caddy	90%	100%
Jetta	69%	100%
Laurent	100%	100%
Rasmussen	0	0
Felix Wankel	0	0
Heinrich Nordhoff	0	0
Ludwig Kraus	0	0
Eberhard Kittler	0	0

Table 6.4 Evaluation Results of Person for GATE

The different types of Named Entities have the different results. Within the types of NE, such as “Organization” and “Location”, we can find the value of recall is better. It is more “relevant” by using GATE to recognize the NE. But the types of NE, such as “Person”, we can find the value of precision is better. It is more “matching” by using it to recognize. We can also find that some “person” NE, GATE can’t recognize, GATE is not perfect for “Person” Named entity recognition. The recall of GATE is almost 100%.

The precision of GATE is about 60%. (According to the average among the whole value.)

### 6.3 Evaluation results of CRFClassifier

The table below will show the results of evaluation.

Organization	Recall	Precision
Audi	98%	29%
Volkswagen Group	100%	57%
NSU	100%	39%
Auto Union	99%	61%
Volkswagen	100%	17%
DTM	100%	61%
Honda	100%	44%
BMW	100%	29%
Toyota	100%	56%
Ford	100%	14%

Table 6.5 Evaluation Results of Organization for CRFClassifier

Location	Recall	Precision
Germany	100%	73%



Hungary	100%	78%
Belgium	100%	100%
China	100%	55%
India	100%	75%
Japan	100%	11%
United Kingdom	100%	78%
Asia	100%	38%
Zwickau	70%	46%
USA	44%	24%

Table 6.6 Evaluation Results of Location for CRFClassifier

Person	Recall	Precision
Johann	100%	20%
Franz	100%	20%
Caddy	50%	10%
Jetta	43%	8.6%
Laurent	0	0
Rasmussen	100%	40%
Felix Wankel	100%	67%
Heinrich Nordhoff	100%	100%
Ludwig Kraus	100%	50%
Eberhard Kittler	100%	100%

Table 6.7 Evaluation Results of Person for CRFClassifier

We can find that CRFClassifier is more suitable for all types of Named entities. The value of recall is better than the results of GATE, but the value of precision is less than the results of GATE. So we can say that NER in CRFClassifier is more relevant than NER in GATE, but it is less matching than NER in GATE. CRFClassifier is used more popular to recognize NE now, such as Wikipedia, it

use CRFClassifier as a standard to identify different kinds of NEs.

The recall of CRFClassifier is 100%.

The precision of CRFClassifier is 70%.

## 6.4 Evaluation results of LbjNerTagger

Because of the form of NER is different from other two systems, we need to transform it to the standard form that the same as the other two systems.

Organization	Recall	Precision
Audi	98%	86%
Volkswagen Group	100%	95%
NSU	98%	63%
Auto Union	96%	87%
Volkswagen	100%	31%
DTM	93%	57%
Honda	100%	22%
BMW	99%	92%
Toyota	100%	72%
Ford	67%	14%

Table 6.8 Evaluation Results of Organization for LbjNerTagger

Location	Recall	Precision
Germany	88%	82%
Hungary	93%	78%
Belgium	100%	80%
China	83%	75%
India	100%	75%

Japan	100%	11%
United Kingdom	100%	100%
Asia	100%	50%
Zwickau	31%	28%
USA	90%	58%

Table 6.9 Evaluation Results of Location for LbjNerTagger

Person	Recall	Precision
Johann	100%	20%
Franz	100%	20%
Caddy	67%	20%
Jetta	74%	57%
Laurent	0	0
Rasmussen	20%	20%
Felix Wankel	100%	100%
Heinrich Nordhoff	100%	100%
Ludwig Kraus	100%	100%
Eberhard Kittler	0	0

Table 6.10 Evaluation Results of Person for LbjNerTagger

We can see that the value of recall and precision of NER in LbjNerTagger is more average than other two systems. No matter the different of types of NER, the results of evaluation reflect that recognized NE by LbjNerTagger is well all round. It is more relevant and matching NER by using LbjNerTagger.

The recall of LbjNerTagger is 100%.

The precision of LbjNerTagger is 70%.

## 6.5 Evaluation Summary

Evaluation by developed program to calculate the value of recall and precision, we can easily find that GATE is more useful for recognize the types of “Organization” and “Location”, CRFClassifier is good at to recognize NE more relevant, but LbjNerTagger is average and can use widely.

Through the process of evaluation, we can find that the function of GATE is more comprehensive. NER function is just one corner of ANNIE, there are several other functions can be used, such as POS Tagger, Sentence Splitter, etc. It is more complex to evaluate NER in such a system. On the other hand, CRFClassifier and LbjNerTagger is more independent, NER is the only function of the systems. We can easily use this function to process the corpus, find the Named Entities. Even so, the types of NE which CRFClassifier and LbjNerTagger supported, is limited. Only “Organization”, “Location” and “Person” are supported to recognize. GATE can support the types of NE not only these, but so many other NE, such as “First Person”, “Date”, etc.

## Chapter 7

## Conclusion

This master thesis has evaluated the existing Named Entity Recognition Systems. This has been part of the ongoing research in the field of information extraction. The objectives of this work were:

1. Research the areas of Named Entity Recognition

Through the research the areas of NER, we can learn the basic and detail definition of NER. We explained when/how NER is used in applications. We list the main challenges of NER systems, and also the benefit of using NER as part of other systems. We also showed historical remarks about NER and detailed discussion of tag sets in NER, explaining in detail what each tag (like Location) mean in terms of grammatical analysis. This master thesis shows the different types of Named Entities, and learning method, feature space of Named Entities.

2. Evaluate existing systems of NER

According to the theory and developed program, this thesis evaluates three existing systems of NER. This procedure contains two parts. Process and give the results of the corpus by existing NER systems and evaluate the results of

them by developed program. Through the evaluation results, we can easily find that GATE is more diversification, CRFClassifier is more standardization and LbjNerTagger is middle of the road. Although these three systems are imperfection, we can also find that GATE can be used more areas not only in Named Entity recognition, but more areas of Natural Language Processing. It also is a “developer” and “embedded” system, so it is very easy to add plug-in to perfect this system. CRFClassifier is used as a standard NER systems by worldwide, such as Wikipedia. LbjNerTagger is designed as a part of project based on NLP, its function is not good enough to be a standard, but still has many referential experiences.

The method of evaluation has limitation. It can't evaluate the precise value of recall and precision. It only can let the people know the main idea of each system. Because we only evaluate the iconic Named entities, it can't show the accurate value of them. We should find a new better evaluate method to judge the advantage and disadvantage of each systems in the future.

## Chapter 8

### Future Work

Through the evaluation of existing NER systems, we can find that it is imperative to develop a perfect system in the future. Although there are several NER systems in the world, these three systems can stand for the most of them, still have some special NER technologies had been researched and developed. We can research the detailed of them in the future.

In the thesis, we only evaluate the iconic Named entities for the existing systems, it is better to evaluate the all NEs. But it is hard to find the right way to definition the whole right Named entities in a huge corpus, cause only the artificial definition should be acceptant to as the standard NEs compare with the NEs that recognized by existing system. CRFClassifier is generally accepted system in the world now. a better system should be developed in the future.

As I discussed before in this master thesis, English has been research very well. Even Chinese, Germany, Japanese and so many other languages has been present as the field of NER, but Norwegian has not research very well. How to find the right way to deliberate Named Entity Recognition system for Norwegian can be consideration in the future.





# Appendix A: Acronyms and Abbreviations

**IDI** Department of Computer and Information Science

**NTNU** Norwegian University of Science and Technology

**NER** Named Entity Recognition

**GATE** General Architecture for Text Engineering

**CRF** Conditional Random Field

**LBJ** Learning Based Java

**ANNIE** A Nearly-New Information Extraction System

**POS** Parts of Speech

**LSA** Latent Semantic Analysis

**NLP** Nature Language Processing

**IE** Information Extraction

**IR** Information Retrieval

**HMM** Hidden Markov Models

**MEMM** Maximum Entropy Markov Model

**SVM** Support Vector Machines

**MUC** Message Understanding Conference



## Appendix B: Digital Appendix

Attached to this report is a zip-file containing the following:

- Source code of the developed program which used to evaluate the existing system
- Original Corpus
- Processed corpus by three existing systems



## Reference

- [1] Neches R, Fikes R E, Gruber T R. Enabling technology for knowledge sharing. AI Magazine, 1991, 12(3): 36~56
- [2] Gruber T R . A translation approach to portable ontology specifications. Knowledge Acquisition, 1993, 5(2): 199~ 220
- [3]Berners-Lee, Tim; Hendler, James; Lassila, Ora (17 May 2001). "The Semantic Web". Scientific American.
- [4] A. Maedche and S. Staab. Learning ontologies for the semantic web. In Semantic Web Worskhop 2001
- [5] Bruckschen et al. on Named Entity Recognition in the Legal Domain for Ontology Population
- [6] NicolasWeber, Paul Buitelaar Web-based Ontology Learning with ISOLDE
- [7] Roman Klinger et al. Named Entity Recognition with Combinations of Conditional Random Fields
- [8] Chieu, Hai Leong and Ng, Hwee Tou, Named Entity Recognition: A Maximum Entropy Approach Using Global Information, COLING'02: Proceedings of the 19th international conference on Computational linguistics, 2002
- [9] Grishman and Sundheim, Message Understanding Conference-6: a brief history, International Conference On Computational Linguistics, Proceedings of the 16th conference on Computational linguistics,1996

- [10] S. Sekine, K. Sudo, C. Nobata: "Extended Named Entity Hierarchy", LREC 2002
- [11] R.Gaizauskas, P.Rodgers, H.Cunningham, and K.Humphreys. GATE User Guide,1996.
- [12] L. Ratinov and D. Roth,Design Challenges and Misconceptions in Named Entity Recognition, CoNLL 2009
- [13] Nicholas Delmonico Rizzolo, An Introduction to Learning Based Java, 2007 195-211. [http://www.linguateca.pt/HAREM/actas/Capitulo\\_11-MotaSantos2008.pdf](http://www.linguateca.pt/HAREM/actas/Capitulo_11-MotaSantos2008.pdf)
- [14] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling.*Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, pp. 363-370.<http://nlp.stanford.edu/~manning/papers/gibbscrf3.pdf>
- [15] D.Maynard, V. Tablan and H. Cunningham. NE recognition without training data on a language you don't speak. ACL Workshop on Multilingual and Mixed-language Named Entity Recognition: Combining Statistical and Symbolic Models, Sapporo, Japan, 2003.
- [16] K. Bontcheva, M. Dimitrov, D. Maynard, V. Tablan, H. Cunningham. Shallow Methods for Named Entity Coreference Resolution. Chaines de references et resolveurs d'anaphores, workshop TALN 2002, Nancy, France, 2002
- [17] D. Maynard, H.Cunningham, R.Gaizauskas. Named Entity Recognition at Sheffield University. Nordic Language Technology -- Arbog for Nordisk Sprogteknologisk Forskningsprogram 2002-2004, 141-145, Museum Tusculanums Forlag. Maynard, V. Tablan, H. Cunningham, C. Ursu, H. Saggion, K. Bontcheva and Y. Wilks. Architectural Elements of Language Engineering Robustness. Journal of Natural Language Engineering -- Special Issue on Robust Methods in Analysis of Natural Language Data, Volume 8, Number 2-3, 257-274, 2002.
- [18] K. Pastra, D. Maynard, H. Cunningham, O. Hamza, Y. Wilks. How feasible is the reuse of grammars for Named Entity Recognition? Language Resources and Evaluation Conference (LREC'2002), 2002

- [19] Shipra Dingare, Malvina Nissim, Jenny Finkel, Claire Grover, and Christopher D. Manning. 2004. A System For Identifying Named Entities in Biomedical Text: How Results From Two Evaluations Reflect on Both the System and the Evaluations. *Comparative and Functional Genomics* 6:77-85.
- [20] Shipra Dingare, Jenny Finkel, Malvina Nissim, Christopher Manning, and Claire Grover. 2004. A System For Identifying Named Entities in Biomedical Text: How Results From Two Evaluations Reflect on Both the System and the Evaluations. In *The 2004 BioLink meeting: Linking Literature, Information and Knowledge for Biology at ISMB 2004*.
- [21] Jenny Finkel, Shipra Dingare, Huy Nguyen, Malvina Nissim, Christopher Manning, and Gail Sinclair. 2004. Exploiting Context for Biomedical Entity Recognition: From Syntax to the Web. *Joint Workshop on Natural Language Processing in Biomedicine and its Applications at Coling 2004*.
- [22] David Nadeau, Satoshi Sekine, A survey of named entity recognition and classification
- [23] Sekine, Satoshi; Nobata, C. 2004. Definition, Dictionaries and Tagger for Extended Named Entity Hierarchy. In *Proc. Conference on Language Resources and Evaluation*.
- [24] Bikel, Daniel M.; Miller, S.; Schwartz, R.; Weischedel, R. 1997. Nymble: a High-Performance Learning Name-finder. In *Proc. Conference on Applied Natural Language Processing*.
- [25] Sekine, Satoshi. 1998. Nyu: Description of the Japanese NE System Used For Met-2. In *Proc. Message Understanding Conference*.
- [26] Borthwick, Andrew; Sterling, J.; Agichtein, E.; Grishman, R. 1998. NYU: Description of the MENE Named Entity System as used in MUC-7. In *Proc. Seventh Message Understanding Conference*.
- [27] Asahara, Masayuki; Matsumoto, Y. 2003. Japanese Named Entity Extraction with Redundant Morphological Analysis. In *Proc. Human Language Technology conference - North American chapter of the Association for Computational*

Linguistics.

- [28] McCallum, Andrew; Li, W. 2003. Early Results for Named Entity Recognition with Conditional Random Fields, Features Induction and Web-Enhanced Lexicons. In Proc. Conference on Computational Natural Language Learning.
- [29] Palmer, David D.; Day, D. S. 1997. A Statistical Profile of the Named Entity Task. In Proc. ACL Conference for Applied Natural Language Processing.
- [30] Mikheev, A.; Moens, M.; Grover, C. 1999. Named Entity Recognition without Gazetteers. In Proc. Conference of European Chapter of the Association for Computational Linguistics
- [31] Whitelaw, Casey; Patrick, J. 2003. Evaluating Corpora for Named Entity Recognition Using Character-Level Features. In Proc. Australian Conference on Artificial Intelligence
- [32] Nadeau, David; Turney, P.; Matwin, S. 2006. Unsupervised Named Entity Recognition: Generating Gazetteers and Resolving Ambiguity. In Proc. Canadian Conference on Artificial Intelligence.
- [33] Brin, Sergey. 1998. Extracting Patterns and Relations from the World Wide Web. In Proc. Conference of Extending Database Technology. Workshop on the Web and Databases.
- [34] Collins, Michael; Singer, Y. 1999. Unsupervised Models for Named Entity Classification. In Proc. of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora.
- [35] Yangarber, Roman; Lin, W.; Grishman, R. 2002. Unsupervised Learning of Generalized Names. In Proc. of International Conference on Computational Linguistics
- [36] Cucerzan, Silviu; Yarowsky, D. 1999. Language Independent Named Entity Recognition Combining Morphological and Contextual Evidence. In Proc. Joint Sigdat Conference on Empirical Methods in Natural Language Processing and Very Large Corpora.
- [37] Riloff, Ellen; Jones, R 1999. Learning Dictionaries for Information Extraction



using Multi-level Bootstrapping. In Proc. National Conference on Artificial Intelligence.

[38] Pasca, Marius; Lin, D.; Bigham, J.; Lifchits, A.; Jain, A. 2006. Organizing and Searching the World Wide Web of Facts—Step One: The One-Million Fact Extraction Challenge. In Proc. National Conference on Artificial Intelligence.

[39] Heng, Ji; Grishman, R. 2006. Data Selection in Semi-supervised Learning for Name Tagging. In Proc. joint conference of the International Committee on Computational Linguistics and the Association for Computational Linguistics. Information Extraction beyond the Document.

[40] Alfonseca, Enrique; Manandhar, S. 2002. An Unsupervised Method for General Named Entity Recognition and Automated Concept Discovery. In Proc. International Conference on General WordNet

[41] Shinyama, Yusuke; Sekine, S. 2004. Named Entity Discovery Using Comparable News Articles. In Proc. International Conference on Computational Linguistics

[42] Rau, Lisa F. 1991. Extracting Company Names from Text. In Proc. Conference on Artificial Intelligence Applications of IEEE.

[43] Wang, Liang-Jyh; Li, W.-C.; Chang, C.-H. 1992. Recognizing Unregistered Names for Mandarin Word Identification. In Proc. International Conference on Computational Linguistics.

[44] Petasis, Georgios; Vichot, F.; Wolinski, F.; Paliouras, G.; Karkaletsis, V.; Spyropoulos, C. D. 2001. Using Machine Learning to Maintain Rule-based Named-Entity Recognition and Classification Systems. In Proc. Conference of Association for Computational Linguistics

[45] Boutsis, Sotiris; Demiros, I.; Giouli, V.; Liakata, M.; Papageorgiou, H.; Piperidis, S. 2000. A System for Recognition of Named Entities in Greek. In Proc. International Conference on Natural Language Processing

[46] Black, William J.; Rinaldi, F.; Mowatt, D. 1998. Facile: Description of the NE System used for Muc-7. In Proc. Message Understanding Conference.

[47] Patrick, Jon; Whitelaw, C.; Munro, R. 2002. SLINERC: The Sydney

Language-Independent Named Entity Recogniser and Classifier. In Proc. Conference on Natural Language Learning.

[48] Da Silva, Joaquim Ferreira; Kozareva, Z.; Lopes, G. P. 2004. Cluster Analysis and Classification of Named Entities. In Proc. Conference on Language Resources and Evaluation

[49] Kripke, Saul. 1982. Naming and Necessity. Boston: Harvard University Press.

[50] Fleischman, Michael. 2001. Automated Subcategorization of Named Entities. In Proc. Conference of the European Chapter of Association for Computational Linguistic.

[51] Fleischman, Michael; Hovy. E. 2002. Fine Grained Classification of Named Entities. In Proc. Conference on Computational Linguistics.

[52] Bodenreider, Olivier; Zweigenbaum, P. 2000. Identifying Proper Names in Parallel Medical Terminologies. Stud Health Technol Inform 77:443-447, Amsterdam: IOS Press.

[53] Ferro, Lisa; Gerber, L.; Mani, I.; Sundheim, B.; Wilson G. 2005. TIDES 2005 Standard for the Annotation of Temporal Expressions. The MITRE Corporation.

[54] Ohta, Tomoko; Tateisi, Y.; Kim, J.; Mima, H.; Tsujii, J. 2002. The GENIA Corpus: An Annotated Research Abstract Corpus in Molecular Biology Domain. In Proc. Human Language Technology Conference.

[55] Tsuruoka, Yoshimasa; Tsujii, J. 2003. Boosting Precision and Recall of Dictionary-Based Protein Name Recognition. In Proc. Conference of Association for Computational Linguistics. Natural Language Processing in Biomedicine.

[56] Rindfleisch, Thomas C.; Tanabe, L.; Weinstein, J. N. 2000. EDGAR: Extraction of Drugs, Genes and Relations from the Biomedical Literature. In Proc. Pacific Symposium on Biocomputing.

[57] Narayanaswamy, Meenakshi; Ravikumar K. E.; Vijay-Shanker K. 2003. A Biological Named Entity Recognizer. In Proc. Pacific Symposium on Biocomputing.

[58] Maynard, Diana; Tablan, V.; Ursu, C.; Cunningham, H.; Wilks, Y. 2001. Named

Entity Recognition from Diverse Text Types. In Proc. Recent Advances in Natural Language Processing

[59] Minkov, Einat; Wang, R.; Cohen, W. 2005. Extracting Personal Names from Email: Applying Named Entity Recognition to Informal Text. In Proc. Human Language Technology and Conference Conference on Empirical Methods in Natural Language Processing.

[60] Poibeau, Thierry; Kosseim, L. 2001. Proper Name Extraction from Non-Journalistic Texts. In Proc. Computational Linguistics in the Netherlands.